Highlights

An Invariant-based Deep Convolutional Neural Network for Real-Time Process Monitoring and Detecting Stealthy Attacks in Water Treatment Plants

Gauthama Raman MR, Aditya P Mathur

- A distributed attack detection framework using invariants based deep convolutional neural network (I-DCNN) is proposed
- I-DCNN integrates the merits of design & data centric approach for effective detection of stealthy attacks on ICS
- Experiments were conducted on realistic water treatment plant namely SWaT testbed by launching single & multi-point coordinated attacks
- We analyse performance of I-DCNN integrated with several statistical techniques in terms of their fastness in detection stealthy attacks.

An Invariant-based Deep Convolutional Neural Network for Real-Time Process Monitoring and Detecting Stealthy Attacks in Water Treatment Plants

Gauthama Raman MR[a,*], Aditya P Mathur[b]

[a] iTrust-Centre for research in cyber security, Singapore University of Technology and Design
[b] iTrust-Centre for research in Cyber Security, Singapore University of Technology and Design, Singapore, and Department of Computer Science, Purdue University, USA

ARTICLE INFO

Keywords:
Industrial control systems
Anomaly detection
Stealthy attacks
Invariants
Convolutional Neural Networks

ABSTRACT

Requesting Prof to write

1. Introduction

Industrial control systems (ICSs) are the major part of several critical infrastructures like water treatment & distribution plants, oil refineries, nuclear power plants, large scale communication systems etc. An ICS combines numerous physical and control components along with its associated communication network to automate the industrial process control. Typically, as shown in Figure 1, ICS consists of devices and subsystems includes sensors, actuators, Programmable logic controllers (PLCs), Human Machine Interference (HMIs), Supervisory Control and Data Acquisition (SCADA) system etc. Generally, the sensors measure the physical entities and convert them to signals for sending it to controllers. These controllers will process the data from sensors and issues the control commands to actuators to achieve the industrial objectives. Overall, the sensors, actuators, controllers along with HMIs are the major constituents of typical ICS. However, in real-time deployment, since ICS is geographically dispersed asset, other control components like SCADA system, PLCs, distributed control system (DCS) are included and configured to achieve a closed loop control with minimal human involvement [29]. For details regarding the process control by the above-mentioned components, the readers can refer [42].

Traditionally, ICS are primarily designed as isolated systems in "air gapped" environment focusing on system functionalities. However, over a period such design has become difficult to deploy, maintain, and supervise. In recent years, the realization of benefits from Internet of Things (IoTs) has strengthened the connectivity between ICS and internet technology for numerous operational gains like scalability, cost-efficiency, remote monitoring, gathering critical information for corporate level decision making etc [42]. On other hand, the growing openness has made ICSs vulnerable to a wide range of insider and outsider threats. For example, the remote access to ICS for monitoring and regulating day-to-day operating has exposed ICS devices to several cyber threats

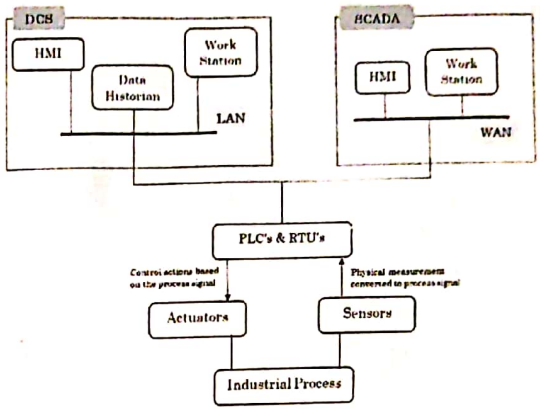

Figure 1: High level view of ICS

common to IT systems like Distributed Denial of Service (DDOS), Advanced Persistent Threats (APTs), ransomware, botnet. Unlike IT infrastructure, the failure of ICS due to cyber-attacks has a serious impact on industrial production, environment safety and even cause direct loss to human life. Thus the design of secured ICSs has become an important topic of research among academic and industrial experts [42, 5].

Motivation: Two reasons that motivated this work as follows. Initially, the existing defence mechanism like Intrusion Detection Systems (IDSs), firewalls, VPNs, other encryption techniques have failed to safeguard the ICSs which is evident from cyber-attack incidents reported in last decade [7, 8]. For instance, Stuxnet worm [21] infecting Iranian nuclear power plant, Black energy malware [23] against Ukraine's power grid, Slammer worm [35] crashing Ohio's Davis-Besse nuclear power plant, etc. Hence, these incidents aggravate the need for ICS specific anomaly detection technique. Second fact is the approach in designing the anomaly detector. Generally, the process flow of ICSs is basically evolved by obeying the fundamentals laws of nature [5]. Since most of the cyber-attack targets the physical components by altering the sensor measurements and control signals, it's enough

*Corresponding author
gauthama_mani@sutd.edu.sg (G.R. MR); aditya_mathur@sutd.edu.sg (A.P. Mathur)
ORCID(s):

ICSs and proposed an architecture for SCADA specific security solutions. Among the existing state of art approaches reported in [11], process based detection methods are predominantly used for anomaly detection in ICSs. In this section, we present some literature on the above-mentioned method using design & data centric approaches.

2.1. Design-centric approaches

Design centric or specification-based models are most commonly used approaches for detecting process anomalies in ICSs. These approaches require a detailed knowledge regarding the complex physical laws that governs the system operation and specification of each components in ICSs. They generally rely on formal mathematical modelling (invariants) of physical process and any behaviour that violates such invariants is termed as anomalies.

Gamage [12] proposed a generalized framework for safeguarding modern Cyber Physical Systems (CPSs) that unifies the security aspects of cyber and physical components. The proposed approach detects anomalies based on the semantics of information flow and was successfully demonstrated on a DC circuit in a smart grid. Tamal et al. [34] presented a unified knowledge model that extracts invariants from smart grid using the theory of Lyapunov-like functions for anomaly detection. Rosich A et al. [39] explores the invariant set for detecting cyberattacks against controllers. The proposed approach was found to be useful in creating large number of control laws for system operator. In [33] authors proposed a state estimation and detection framework for power networks. In this work, the minimum variance of network state is estimated through the distributed computing for anomaly detection. The above mentioned research contributions majorly focus on the power systems and notably all are theatrical approaches.

The work of Adepu S [5] namely distributed attack detection (DAD) was proposed to detect cyber attacks in real time, through the identification of anomalies in water treatment plants. It derives invariants from the plant design of SWaT testbed and detects anomalies by monitoring entire plant operation or plant in the given state. The effectiveness of DAD was experimentally validated in several case studies [3, 4, 6] by launching real time stealthy and coordinated attacks. As the extension of this work, authors of [11] implemented a similar approach for water distribution plant (WADI).

The design centric approaches generally have a good performance in anomaly detection process of ICSs. However, its significant drawback is the extraction of invariants. Formulation of invariants requires deep knowledge regarding the plant operation which is a time consuming process. Further, these invariants are sensitive to noise due to dynamic nature of ICSs, aging factor of ICS devices, human misconfigurations, and incorrect technical documents. For example, in the case of SWaT testbed (discussed in Section 3.1) unlike specified in the operational manual, opening and closing of motorized value is not immediate. It incurs a time delay of 6 seconds (approx.) to change its state completely. Due to this, the water flow rate is 11% more than the actual value reported in the document. Further, in the vendor documentation, the upper bound of the tank was mentioned as 1100mm and encoded in PLCs. While during the normal operation of plant, water level does not exceeds 830mm. As these operational discrepancies cannot be captured by design-centric approaches, motivates the need of machine learning model (data centric methods) for the detecting anomalies in ICSs.

2.2. Data-centric approaches

Data-centric or data driven approaches integrates several machine learning and statistical models to capture the key behaviour of the system through historical data. Unlike design centric approaches, they are easy to deploy and adaptable to the dynamic nature of ICSs. These approaches are broadly classified into supervised and unsupervised models. The supervised learning models [38, 20, 36, 45, 19] constructs a signature database with known attacks and detects anomalies whenever the plant's state matches with the signature. They generally possess a higher detection rate for known attacks. However, the major drawback in this approach is their inability in detecting zero day vulnerabilities due to lack of signatures. Although frequent updating of signature database paves a better solution, but generation of signatures for process anomaly is a complex task in an multi-variate nature of ICSs.

Due to above mentioned shortcomings and existence of abundant data corresponding to the normal operation of plant, insisted the researchers to consider anomaly detection in ICS as *"one class classification problem"*. Recently, several unsupervised learning models were effectively used for anomaly detection task. The authors of [41], proposed a model based attack detection and mitigation framework for automatic generation control (AGC) systems in power grids. In this work, several regression models, statistical techniques were deployed to predict the load of AGC and malicious data injection attacks are detected by comparing the observed and predicted load. Hu Y et al. [16] designed an anomaly detector based on permutation entropy approach for ICSs. Initially, kalman filter forecasting model is utilized to predict the behaviour of system and compared against the actual measurement for residue generation. Further, using permutation entropy approach, non-randomness in the magnitude of residuals are captured for detecting stealthy attacks. A similar kind of work was carried out using residual skewness analysis for stealthy attacks detection in ICSs, by the authors of [17]. Shalyga D et al. [40] proposed a deep learning based anomaly detector for water treatment plant. Several fine-tuned neural network models like mulitlayer perceptron, CNN, recurrent neural networks were utilized for anomaly detection process and experiments were carried out on SWaT dataset. Kravchik M & Shabtai A [25] performed a comparative study on the performance of 1D-CNN and autoencoder (AE) model for attack detection. These neural networks were applied on both time & frequency domain of three public datasets namely SWaT, WADI, BATADAL for anomaly detection process. From the experimental analysis, it was found that AE model

to monitor the "physics" of system to detect wide range of anomalies. Accordingly, the anomaly detector must build a model by considering the interaction among the physical devices and then detects the anomalies to achieve better detection rate and minimal false alarms.

Problem Context: Generally, an ICS consists of one or more coordinated stages and each stage possess several physical process evolves over a time in accordance to its controller. However, due to physical damage or cyber-attacks these process will move to an undesirable state (abnormal behaviour) termed as *"process anomaly"*. In this work, we mainly focus on the detection of such process anomalies resulted out of cyber-attacks. Further, from our experience, we observe a change in trend from direct attacks to stealthy attacks against ICSs. The skilled attackers possess complete knowledge regarding the system configuration and inject false information into the system over a longer period of time to achieve their goal and remains undetected. Primarily we focus on the detection of such stealthy attacks before achieving its intended goal.

Solution strategy: Generally, the process based anomaly detection techniques in ICSs can be broadly categorized into two, namely (i) design-centric approaches (ii) data-centric approaches. The anomaly detection process in the former case is based on the physical relationship among the ICS components obtained from the plant design, whereas the latter case rely on feature learning concept through the application of machine learning algorithms [5, 13]. Although both approaches has its own pros and cons, Azmi et al. [43] has carried out a research work, stating the need for integrating these two approaches. Taking that into consideration, we design an anomaly detector based on deep convolutional neural network (DCNN) for an operational water treatment plant and analysis its effectiveness in detecting single point & multi point coordinated stealthy attacks.The proposed work consists two phases, wherein initially we study the interaction among the state variables[1] (invariants) from the plant[2] design and transform to a physical model using DCNN. Further, the output of the physical model is compared against the observed measurement and finally the potential anomalies are detected using the statistical analysis. We have also experimentally investigated the performance of DCNN integrated with several statistical approaches, namely Cumulative Sum (CUSUM), Gaussian Distribution (GD), Permutation Entropy (PE), Residual Skewness (RS) in terms of their fastness in the detection of stealthy attacks.

Research Questions: The following research questions are formulated to address the challenges in the integration of design-centric and data-centric approaches for the design of an efficient anomaly detector. **RQ1:** How to effectively model the physical process of ICS using Deep convolutional neural network? **RQ2:** How fast and accurately detect the stealthy attacks before its objective is achieved?, and **RQ3:** Is there a possibility of locating the exact ICS component under threat?

Novelty of the proposed approach: The proposed anomaly detector (I-DCNN) combines the advantages of both design and data-centric approaches ie., physical process of ICS is modelled using the invariants extracted from the plant design and threat detection process is carried out based on the residual error analysis using statistical approach. Unlike other deep learning approaches for detecting process anomalies discussed in [18, 40, 27, 25, 24], the proposed I-DCNN operates the physical relationship among the state variables of both discrete and continuous in nature. Further, a comparative study was performed on DCNN integrated with several statistical approaches to detect abnormality between the estimated and actual measurement from sensor while the plant is operational. Thus the novelty and generality of proposed I-DCNN lies in the methodology which was experimentally evaluated on the industrial strengthen water treatment plant (in contrast to the works mentioned above, carrying out the experiments on simulated environments). To summarize (i) I-DCNN integrates both design and data centric approaches for detection of process anomalies (ii) it effectiveness is experimentally evaluated on a realistic testbed by launching single & multi point coordinated stealthy attacks (iii) possess minimal false alarms.

Contributions: (i) A distributed attack detection mechanism (I-DCNN) using invariant based on deep convolutional neural network is proposed for safeguarding ICSs against cyber and physical attacks (ii) the hyperparameters of DCNN is fine tuned for effective modelling of physical process in ICSs (iii) a comparative study has been carried out using several statistical techniques that make use of randomness contained in the predictive residual sequence for faster and accurate detection of stealthy attacks (iv) the experimental assessment of I-DCNN was conducted on a realistic water treatment testbed.

Organization: The remainder of this paper is organized as follows. Section 2 reviews the recent research on design & data centric approaches for detecting process anomalies in ICS. A brief introduction regarding the SWaT architecture, followed the fundamentals concepts on modelling the process of SWaT through invariants & attack detection strategy is discussed in Section 3. Section 4 provides the mathematical insights of DCNN and several statistical technique utilised in the reported work Section 5 presents the case study to evaluate the effectiveness of I-DCNN followed by details of the experiment conducted and performance analysis of I-DCNN variants is discussed in Section 6. Finally Section ?? offers conclusions from this work with insights gained and proposed future directions for research in anomaly detection in the context of ICS.

2. Related Works

Although there exist several works [22, 32, 42, 10, 2] that provides guidelines, standards, and best practices to counteract the potential vulnerabilities against modern ICSs, still the design of secured CIs is a changeling task. Recently Zhu, B.X. [46] discussed the taxonomy of cyber-attacks against

[1] The term *"state variables"* used here to refer the components of ICS

[2] The term "plant" is used here to refer to a critical infrastructure and includes the ICS and the controlled physical process.

has better performance in terms of precision and similarly 1D-CNN model has a higher recall value. Finally, it was concluded that, ensemble approach that combines these two models working on frequency domain has a better detection coverage for all three datasets. In addition to these contributions, there exist other works utilizing unsupervised learning models like auto regression [15], generative adversarial neural network [27, 28], one class support vector machine [18] etc. for anomaly detection in several ICS testbeds like Tennessee Eastman process(TEP), SWaT, gas oil heating loop process etc.

One notable similarity across the work mentioned above is the usage of models that represent the physical behaviour of physical system. According to Cárdenas et al. [9] these representative models are derived either by the law of first principles(law of physics) or through the application of empirical dataset. Due to lack of design knowledge and non-linear relationship among the state variables, most of the reported works utilize training dataset for model creating. However, from a survey conducted by the authors of [37], has proven the significance and feasibility of designing secured CIs based on the representative model obtained from the first principles. Hence in this work, we propose an effective anomaly detector for operational ICS by integrating design and data centric approaches. The proposed I-DCNN, initially study the interaction among the ICS components from the plant design and their non-linear relationship are modelled by DCNN through the historical dataset. Further, they are integrated with several statistical techniques and their effectiveness in detecting stealthy attacks are experimentally evaluated on operational SWaT testbed.

3. Preliminaries

The anomaly detection process in ICS consists of three major steps. Initially, physical modelling of each process of underlying plant were carried to predict the expected output. Secondly, the expected output of each process is compared with its actual measurement for computing residues. Finally, these residues were analyzed using statistical techniques for detection of potential anomalies. The following section discusses the higher-level overview of fully autonomous, 6-stage water treatment plant named SWaT testbed followed by the physical model of each process in SWaT through invariants and finally the anomaly detection strategy.

3.1. Architecture of SWaT testbed

SWaT testbed is for water treatment, designed by Singapore University of Technology and Design (SUTD) in collaboration with Singapore's Public Utility Board (Nations Water utility company). It is basically built for the research purpose to investigate the response of the operational ICS during the cyber-attacks and facilitates the validation of several defence mechanisms. The physical and cyber portion of SWaT testbed is described below.

Water treatment process: The water treatment process in SWaT testbed consists of six coordinated sub-process referred from P1 through P6. Each sub-process are termed as stages, controlled by independent Programmable Logic Controllers (PLCs). The control action by each PLCs are based on the system state estimated from the sensor values.

As shown in Figure 2, the stage P1 controls the inflow of raw water to be treated by the plant through a motorized valve, connected to a raw water tank (T101). Then the water from the tank (T101) is passed to a chemical dosing station in stage 2 (P2) using a pump (P101) followed by the Ultra-Filtration (UF) feed tank (T301) in stage 3. In this stage, the water is passed to a UF unit for removing undesirable materials and through UF feed pump, the water is send to Reverse Osmosis (RO) feed tank (T401). Through the ultra-violet dichlorination unit, the excess chlorine from the water in T401 is removed and passed through a 2-stage Reverse Osmosis (RO) filtration unit in stage 5. Finally, the filtered water from RO process is stored in a permeate tank of P6 and kept ready for distribution. Further the stage 6 also controls the cleaning of UF membranes in stage 3 through a back-wash process. For every 30 minutes, the cleaning process is initiated by switching "ON" the backwash pump. This process can also be initiated by PLC 3, when the pressure drops across the UF unit measured by differential pressure sensor value exceeds 0.4 bar, indicating the UF membranes are clogged.

Communications: Each stages in SWaT are equipped with sensors like flow meters, level sensor, conductivity and acidity analyzers that continuously monitor the physical and chemical properties of water and report to corresponding PLCs in each stages. Based on the obtained sensor values, the PLCs send the control signals to actuators like pump & motorized valve in their domain. These PLCs will also communicate with each other through a separate network. Communications among sensors, actuators, and PLCs can be via either wired or wireless links; manual switches allow transfer between the wired and wireless modes.

3.2. Physical modelling of ICS

The entire process of large plant is configured into a set of sub-process for easy monitoring and maintenance. Further, these sub-process evolve over a time based on the design of controllers and can be characterized as a physical model for predicting the plant's behavior in future. In this work, we rely on process invariants for model generation. A *"process invariant"*[3] exhibits the mathematical relationship or interaction among the plant's components controlled by one or more PLCs. Let us consider a plant with "n" number of components denoted as $C=\{c_1, c_2 \ldots c_n\}$. Let $D=\{d_1, d_2 \ldots d_T\}$ be a "n" dimensional time series data obtained from the plant at discrete time steps $t \in [1, 2, 3 \ldots T]$. These data logs can be represented as $d_t=\{u_t, v_t\}$, where each element in u_t & v_t are the real value readings obtained from sensors and categorical value corresponding to the state of actuators respectively. During the plant operation, the generated models were utilized to predict the state of the corresponding component and compared against the actual state. Based on the discrepancy between the actual and predicted states, alerts

[3] hereafter will be referred as *"invariants"*

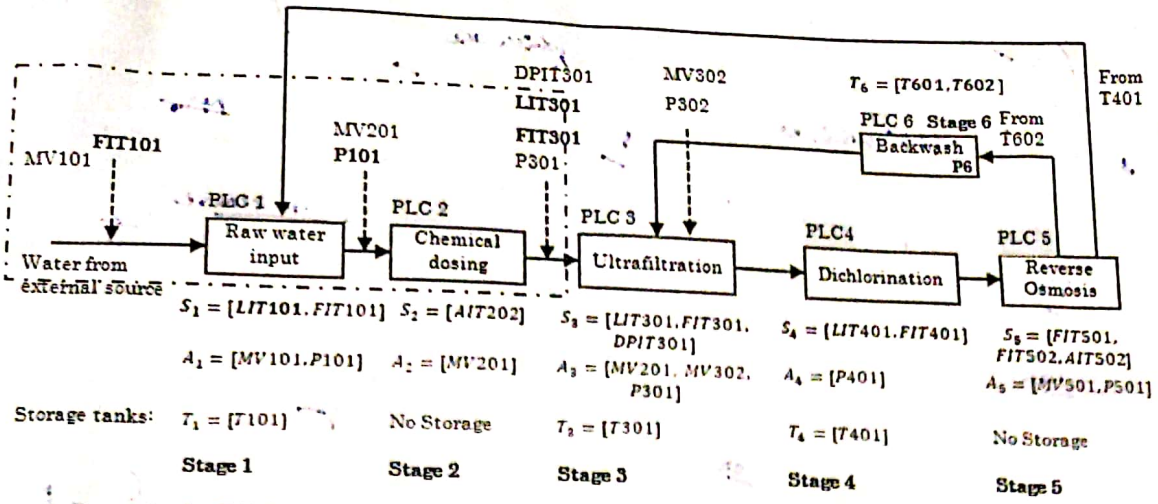

Figure 2: Stages in the SWaT water treatment plant. LITxxx: level sensor; Pxxx: pump; AITxxx: chemical property sensor; FITxxx: flow rate indicator; DPITxxx: differential pressure indicator. Experiments reported in this paper used sensors marked in bold located at stages enclosed by dashed lines.

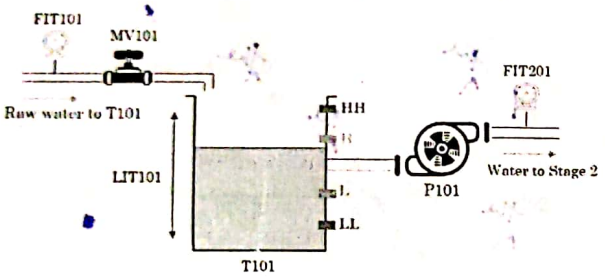

Figure 3: Stage 1 of SWaT testbed

are raised. Mathematically, let *"s"* denote a state variable to a model of any component in C and *"y"* be its measurement. Let us assume a linear relationship between the s & y, that can be defined as follows,

$$s(t+1) = \beta_1 s(t) + \beta_2 \phi(t) \quad (1)$$

$$y(t) = \beta_3 s(t) + \beta_4 \phi(t) + \epsilon(t) \quad (2)$$

Where $\beta_1, \beta_2, \beta_3, \beta_4$ are constants, $\phi(t)$ is the control variable and y(t) represents the sensor measurement of state s(t) at time t. Through the application of predictive algorithms, we can estimate s(t), which is denoted as $\hat{s}(t)$. During the normal operation of plant ie in absence of noise and attack, y(t)=s(t)=$\hat{s}$(t). However, in case of attack, the expected relationship will not be valid stating that sensor is compromised. The following example illustrate the generation of invariants for level sensor (LIT101) of tank T101.

Example 1: As shown in Figure 3, the stage 1 of SWaT testbed consist of three sensors namely LIT101 (measuring the water level of tank T101), FIT101 (measuring the water flow rate into T101) and FIT202 (measuring the water flow rate out of T101). At time *"t"*, let s(t) be the water level of T101 and ϕ_{in} & ϕ_{out} denotes the inflow and outflow of water respectively. Theoretically, the invariants for estimating the water level of T101 at "t+1" as follows,

$$s(t+1) = s(t) + \delta(\phi_{in}(t) - \phi_{out}(t)) \quad (3)$$

where, δ is the constant, computed based on the physical properties of tank and flow rate. For more information regarding the generation of invariants for SWaT tested, the readers can refer [5] . Rearranging the Eqn.(3), we get

$$s(t+1) = s(t) + \delta(\phi_{in}(t) - \phi_{out}(t)) \quad (4)$$

where Eqn.(4) is valid only for ideal sensors. Further, practically deriving the relationship among the state variables for a given invariant is a complex task due to the existence of higher order & non-linear correlation among them. In order to overcome this issue, we use DCNN (Section 4.1) to capture the relationship between the dependent and independent state variables defined in Eqn.(5)

$$s(t+1) = f(s(t), \phi_{in}(t), \phi_{out}(t)) \quad (5)$$

From Eqn.(5) it is clear that for given plant with normal behavior of components and its corresponding invariants, we can build a nonlinear model (f) that can estimate the future behavior of components with minimal error.

3.3. Detection Strategy

Once the physical model of each sub process of the underlying plant is obtained, the forecasting procedure is performed for computing residues. Further through the application of statistical techniques the existence of anomalies is

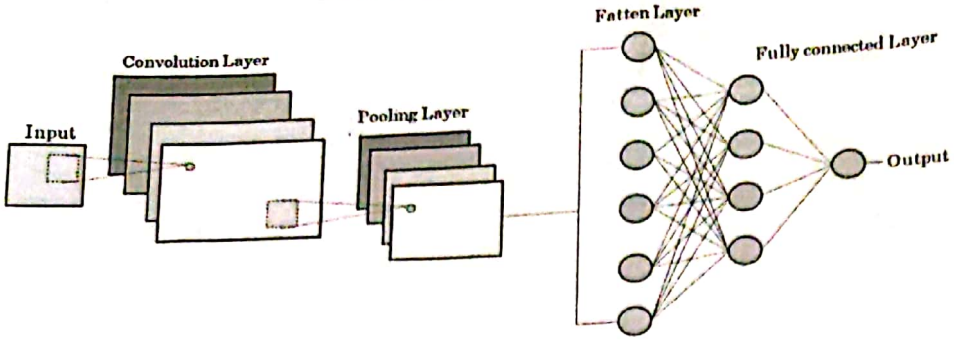

Figure 4: CNN architecture

detected. The core idea is to identify any significant deviation between the observed measurement and the output of the trained model as it can estimate the behavior of components during their normal condition. According to a survey carried out by the authors of [44], there exists two different approaches in analyzing the residues namely, Stateless and Stateful.

In stateless approach, the alter will be raised for each instance, when there is deviation between the actual and prediction value. Mathematically, for $\forall t > 0, if\ |s(t) - (t))| = r(t) > |\tau|$, where τ is the predefined threshold, alert will be raised. Unlike the above approach, the stateful approach will continuously monitor the residues (r(t)) and when the number of abnormal deviations increases beyond the predefined time steps (t_w), the alter will be raised. In this work, we experimentally analysis the performance of several stateless stateful statistical approaches discussed in Section 4.3 that are integrated with DCNN, for detecting real time, stealthy attacks against the SWaT testbed.

4. Detecting Stealthy attacks using I-DCNN

The mathematical insight and working of DCNN along with the impact of hyper-parameter tuning in its performance are discussed in this section. Further, we present several statistical techniques used in this study for effective detection of stealthy attacks against SWaT testbed

4.1. Convolutional Neural Network

The convolutional neural network (CNN) is a special class of feed forward neural network designed by LeCun Y et al. [26] for handling dataset with rasterized features for classification and regression task. A deep CNN (DCNN) has multiple layers that performs linear and non-linear operations for extracting high levels of abstractions from the input data. Unlike traditional neural model, CNN is invariant to scaling, shifting and distortion of input data due to key factors like sparse connectivity, parameter sharing, sub sampling and local receptive fields [31]. Each data in CNN is visualized in a form of 2D (2-dimensional)array, thereby avoiding complex feature extraction and reconstruction process. The typical architecture of CNN consisting of convolutional layer, pooling layer and fully-connected layers is depicted in Figure 4. The functional description of each layers as follows [30]:

1. **Convolutional layer**: This layer performs the convolution operation (Eqn.(6) between the input data and linear filter for the generation of feature maps.

$$[h_k]_{mn} = \varphi((W_k \circledast x)_{mn} + b_k) \tag{6}$$

where x is the input data, $[h_k]_{mn}$ is the $(mn)^{th}$ element of k^{th} feature map, $\circledast$ represent the 2D convolution operation, φ is the activation function, $W_k \& b_k$ are the weight and bias of k^{th} convolutional filter.

2. **Pooling layer**: Upon the completion of convolution process, k feature maps are given as input to the pooling layer. During the pooling process, the dimensions of feature maps are reduced to minimize the computational complexity of the network. There are two commonly used pooling operations in CNN namely maximum pooling and average pooling. As shown in Figure 5, the maximum pooling process returns the element with maximum value and similarly the average pooling process provides the average of all element. The maximum and average pooling can be computed as follows

$$[p]_{mn} = A_P([h_k]_{mn}) \tag{7}$$

$$[p]_{mn} = M_P([h_k]_{mn}) \tag{8}$$

where $[p]_{mn}$ represents the output of pooling operation, $A_P \& M_P$ corresponds to the average maximum pooling operation respectively

3. **Flatten & Fully-connected layer**: The role of the flatten layer is to flatten the 2D data to 1D vector representation for fully-connected layer (Figure 6). The fully-connected layer consists of neurons connected

and evaluating the magnitude of the residues generated during the operational plant for anomaly detection process.

1. **Cumulative SUM (CUSUM)**
Cumulative SUM (CUSUM) method is the most commonly used stateful approach for detecting abnormal deviations that corresponds to process anomalies. It computes the cumulative sum of the residual sequence to detect minor deviations in their magnitude. In order to minimize the false positives, we calculate both positive and negative side deviation of residues using Eqn.(12) & Eqn.(13) respectively.

$$P(t) = \max(0, r(t) - target - b); t = 1, 2, .., T \quad (12)$$

$$N(t) = \min(0, r(t) - target + b); t = 1, 2, .., T \quad (13)$$

where P(t) & N(t) calculates the high & low cumulative sum respectively. target represents the safety limit and b is the allowable slack. Further, we introduce two parameters namely Upper Control Limit (UCL) & Lower Control Limit (LCL), which are computed empirically using the values of P & N. During the testing phase, when P(t)>UCL or N(t)<LCL, then behaviour of the component is termed a "suspicious".

2. **Permutation Entropy (PE) approach**: Permutation Entropy approach has been extensively used in the identification of non-randomness in the time series data, using which the anomaly detection process is carried out. For a residual sequence $\{r(t)\}_{t=1,2..T}$, we generate a phase-space reconstruction matrix defined in Eqn. 14

$$P_s = \begin{pmatrix} r(1) & r(1+\xi) & \dots & r(1+(\lambda-1)\xi) \\ \vdots & \vdots & \dots & \vdots \\ r(i) & r(i+\xi) & \dots & r(1+(\lambda-1)\xi) \\ \vdots & \vdots & \dots & \vdots \\ r(K) & r(K+\xi) & \dots & r(K+(\lambda-1)\xi) \end{pmatrix} \quad (14)$$

where $1 \le i \le K, K = T - (\lambda - 1)\xi)$, λ is the order of permutation entropy and ξ is the time delay. Each row in matrix $(P_s(i))$ meets the permutation pattern $\{\pi_j | 1 \le j \le \lambda!\}$ with $\lambda!$ possible permutation. Therefore, the permutation entropy $(H(\lambda))$ is defined as follows.

$$H(\lambda) = -\sum_{j=1}^{\lambda!} p(\pi_j) \log p(\pi_j), \quad (15)$$

and $p(\pi_j)$ is given in Eqn.(15).

$$p(\pi_j) = \frac{L_i}{T - \lambda + 1} \quad (16)$$

where L_i corresponds to number of $P_s(i)$ matching the permutation pattern π_j. During the attack free scenario, the permutation entropy will be relatively larger due to the randomness of residual sequence. However, in the case of anomalies, there exists some sort of regularities in the residual sequence and thereby the permutation entropy decreases. Thus in the testing phase, if the permutation entropy value decreases steeply over a period of time, implies attack is detected and alert has to be raised.

3. **Residual Skewness (RS) approach**: Residual skewness (RS) approach is a stateless detection technique that make use of skewness contained in the predicted residual sequence for anomaly detection. Generally the value of residual skewness coefficient (Eqn.(17)) can distinguish the residues generated during the attack state from residues obtained during the normal state of plant's operation.

$$R_c = \frac{\sum_{i=1}^{l}(r_i - r_m)}{r_s^3}. \quad (17)$$

where l is the length of the sequence, r_m & r_s are the mean & standard deviation of the residual sequence. During the testing phase, when R_c is greater than the predefined threshold ie $R_c > |\tau|$, proves the existence of anomalies.

4. **Gaussian Distribution approach (GD)**
Gaussian Distribution approach (GD) is a stateless detection technique, where the existence of anomalies are identified through the application of univariate normal distribution process. In this approach, the attack free residual series are modelled to fit the Gaussian distribution as defined in Eqn.18. For a given residual sequence r(t), where $r \in \mathbb{R}$, the mathematical expectation (μ) and variance (σ^2) are shown in Eqn.19 & Eqn.20

$$r(t) \sim \mathcal{N}(\mu, \sigma^2) \quad (18)$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} r(t) \quad (19)$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(r(t) - \mu)^2 \quad (20)$$

Its probability density function (p.d.f) is defined as follows,

$$p(r; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(r-\mu)^2}{2\sigma^2}\right) \quad (21)$$

In the testing phase, the classification is done based on the p.d.f value. If the computed p value of the operational residual sequence is less than the user-defined threshold ie $p(r) < \tau$, then the data point is consider as anomaly.

5. Case study

The effectiveness of I-DCNN was assessed experimentally through a case study by designing stealthy attacks against SWaT, launching them, and observing the plant behaviour. In this section, we discuss about the attacker model, attack design and its launch procedure in SWaT testbed.

5.1. Attacking SWaT testbed

The communication infrastructure of ICS are generally connected to the external network for easy monitoring and controlling, making it susceptible to cyber attacks. In current design of SWaT tested, although it is not connected to any external network but there exists a wired & wireless network that connects PLCs, sensors, actuators, HMI & SCADA servers etc. Cyber-attacks exploiting the vulnerabilities in the network protocols, PLC firmware would have the possibilities to compromise the communication link between the sensor & PLCs, actuators & PLCs or among the PLCs. With one or more compromised links, the attacker can manipulate or send the fake data to PLCs causing huge impact on plant's productivity or even damage to physical components.

5.2. Attack models & attack design

A cyber attack can be referred as a sequence of action applied on a operational plant with an objective to move the plant to an anomalous state. Thus the attacker model can be represented as a three tuple vector (A_T, A_P, A_G), where A_T correspond to the type of attack launched through a finite set of points (A_P) with an objective (A_G). An objective (A_G) of the attacker can be represented as a statement. For example, damage the valve (MV101) in water treatment plant, alter the pH value of water coming from distribution plant. Further, an attack is consider to be a single point, if only one point is utilized for launching it ie $A_P = 1$. As there exist several attack types in specific to the considered plant discussed in [5], for a proof of concept, we have taken only stage 1-3 for experimentation. The stealthy attacks with specific intention are launched against the components of these stages to evaluate the performance of I-DCNN.

SWaT testbed is designed to operate 24x7 without any interruption. As discussed in Section 3.1, the raw water from the tank T101 is passed to tank T301 for Ultra-Filtration. Each tanks has four water level makers, namely LL (Low Low), L (Low), H (High), HH (High High) and water levels are monitored by the sensor "LIT101" and "LIT301" and reported to their corresponding PLCs. In order to maintain the normal plant operation, the water level in the both tanks are always maintained above or at level "L". The motorized valve (MV101) is turned ON by PLC1 whenever the water level in T101 goes below "L", letting raw water to flow in. Similarly, when the water level of T301 is at "L", then pump P101 is turned "ON" by PLC3, thereby water from T101 flows to T301. Further stage 4-6 operates continuously for water filtration. Let us consider the normal operation of plant is affected by an attacker through injecting false sensor measurements to PLCs. The intention of the attacker could be minimizing the production rate or even shutting down the plant fully. These attacks are both single point & multi-point in nature, and Table 1 list out the different types of attacks along with its intention & launch procedure, considered for the reported study.

The success of an attacker in realizing its objective and remain undetected depends on two factors (i) technical knowledge of attacker in designing and launching stealthy attacks (ii) strength of the defence mechanism monitoring the plant. For example, consider an attacker with an intention to overflow the tank T101. Thus LIT101 will be its target and single point attack is launched. For each time instance after LIT101 increases above 750mm, the attacker will reduce its value and send it to PLC1, such that the difference between actual and predicted measurement will be Δ. As shown in Figure 8, there exists different cases with varying value of Δ such that, $\Delta_1 > \Delta_2 > \Delta_3$. From the Figure 8(a), it evident that the attacker lacks in the information regarding the components specification & existing defence mechanisms of plant since there exist a maximum deviation from the actual measurement. Due to that, before attaining the intended objective, the attack will be detected (point "B" in Figure 8(a)). Similarly, if we consider the value of Δ_2, although it cannot be detected as faster than the previous case, it will be detected later at point "B" (Figure 8(b)) but still the objective is achieved.

In the final case, Δ_1 with minimum magnitude implies the attacker has a complete knowledge regarding the process flow of the plant and also its defence mechanism, thereby its objective will be achieved at the point "A" in Figure 8(c) and remains undetected. As discussed in Section 4.3, the traditional defence mechanisms operating over the magnitude of residues will fail to detect the stealthy attacks. Hence, let us assume the attacker is of insider type, holding the access for certain physical and cyber components of SWaT testbed as depicted in Figure 9. Further, attacker is also aware of specification and dimension of each SWaT components along with the existing defence mechanism, so that their values are computed and altered in real time during the attack launch.

5.3. Launch procedure

Prior to launch of stealthy attacks, the plant was operated and brought to steady state. No attacks were launched during the plant in transient state ie, soon after it starts. In order to ensure the plant is operating exactly in the steady state, we monitor the gallons/minute of pure water produced at different time instance. Initially, the plant was operated in normal state for more than one hour and then a series of attacks were launched. In order to avoid the cascading effect of one attack on another, a gap of 10 minutes is maintained between the successive attacks. Further each attack were launched five different times at various steady state to analyze the exact response of the proposed I-DCNN. The general procedure followed during the launch of stealthy attacks as follows.

1. Identify the memory location (tag) in PLC where the measurement of the targeted sensor will be saved
2. Compromise the wireless link between the PLC and SCADA workstation

Table 1
Attacks launched against sensors in stage 1-3
(*Single point attack;**Multi-point attack; H=800mm,L=500mm; X=[5mm-10mm]; Y=[0.1cm\hr-0.5 cm\hr])

Attack Reference	Targeted sensor	Objective	How it is launched?
A1*	LIT101	Overflow the tank T101	Water level of tank T101 is about to reach "H"; Attacker takes the control of point "C" in Figure 9 to reduce the LIT101 value by 10mm and sent to PLC1; PLC1 assumes the water level did not reach "H" and does not close MV101.
A2*		Underflow the tank T101	Water level of tank T101 is about to reach "L"; Attacker takes the control of point "C" in Figure 9 to increase LIT101 value by 20mm and sent to PLC1; PLC1 assumes the water level is not below "L" and does not open MV101.
A3*		Overflow the tank T101	Water level of tank T101 is about to reach "H"; Attacker takes the control of point "C" in Figure 9 to reduce the LIT101 value by "X" mm for every second and sent to PLC1; PLC1 assumes the water level did not reach "H" and does not close MV101.
A4*		Underflow the tank T101	Water level of tank T101 is about to reach "L+50mm"; Attacker takes control of point "C" in Figure 9 to increase LIT101 value by "X"mm for every second and sent to PLC1; PLC1 assumes the water level did not reach "L" and does not open MV101.
A5*	FIT101	Slow filling of T101	Flow rate to tank T101 is about to reach 2 cm/hr; Attacker takes control of point "A" in Figure 9 to send a false value of 1.5 cm/hr is sent to PLC1.
A6*			Flow rate to tank T101 is about to reach 1.5 cm/hr; Attacker takes the access of point "C" in Figure 9 and FIT101 is reduced by "Y" cm\hr for every second.
A7*	LIT301	Overflow in T301	Water level of tank T301 is about to reach "H"; Attacker takes the control of point "F" in Figure 9 to reduce the LIT301 value by 20mm and sent to PLC3; PLC3 assumes the water level did not reach "H" and does not close P101.
A8*		Underflow in T301	Water level of tank T301 is about to reach "L"; Attacker takes the control of point "F" in Figure 9 to increase LIT301 value by 20mm and sent to PLC3; PLC3 assumes the water level is not below "L" and does not open P101.
A9*		Overflow in T301	Water level of tank T301 is about to reach "H"; Attacker takes the control of point "F" in Figure 9 to reduce the LIT301 value by "X" mm for every second and sent to PLC3; PLC3 assumes the water level did not reach "H" and does not close P101.
A10*		Underflow in T103	Water level of tank T301 is about to reach "L+50mm"; Attacker takes control of point "F" in Figure 9 to increase LIT301 value by "X" mm for every second and sent to PLC3; PLC1 assumes, the water level did not reach "L" and does not open P101.
A11*	FIT201	Affecting chemical dosing process	Flow rate to tank T301 is about to reach 2 cm/hr; Attacker takes the control of point "E" in Figure 9 and reduce the value of FIT201 to 1.5 cm/hr.
A12*			Flow rate to tank T301 reaches 1.5 cm/hr; Attacker takes the access of point "E" in Figure 9 and FIT201 is reduced by "Y" cm\hr for each second
A13**	LIT101, FIT101	Overflow the T101	Water level of tank T101 is about to reach "H-50mm"; Attacker takes control of point "C" & "A" in Figure 9 and for each timestep, LIT101 & FIT101 is reduced by "X" mm & "Y" cm\hr respectively.
A14**		Underflow the T101	Water level of tank T101 is about to reach "L+50mm"; Attacker takes control of point "C" & "A" in Figure 9 and for each timestep, LIT101 & FIT101 is increased by "X" mm & "Y" cm\hr respectively.
A15**	LIT301, FIT201	Overflow the T301	Water level of tank T301 is about to reach "H-50mm"; Attacker takes control of point "F" & "E" in Figure 9 and for each timestep, LIT301 & FIT201 is reduced by "X" mm & "Y" cm\hr respectively.
A16**		Underflow the T301	Water level of tank T301 is about to reach "L+50mm"; Attacker takes control of point "F" & "E" in Figure 9and for each timestep, LIT301 & FIT201 is increased "X" mm & "Y" cm\hr respectively.

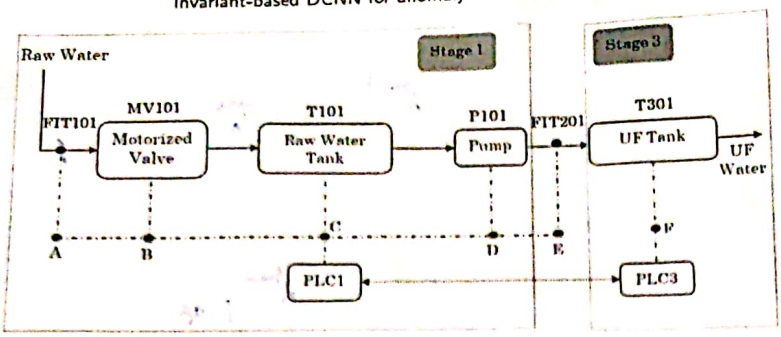

Figure 9: SWaT components involved in the attack and detection scenarios

3. Change the tag to a different value, so that PLCs assumes the manipulated value to be a true state of the component that leads to wrong control decisions.

6. Results and discussions

This section demonstrates the performance of I-DCNN, integrated with several statistical techniques through a case study (discussed in Section 5) in terms of accurate and faster detection of stealthy attacks. The remainder of this section describes the experimental setup, data collection and techniques used to fine tune the hyperparameters of DCNN.

6.1. Experimental setup

The proposed I-DCNN for SWaT testbed was implemented in Python 3.7 using Keras deep learning library. The entire experimentation was carried out in Intel Xeon processor running in Windows 7 OS with 64GB RAM. For training and validation purposes, we have utilized the records corresponding to normal operation of plant obtained from SWaT tested dataset [14]. The testing process of the proposed anomaly detector was carried out on the data from operational SWaT testbed. An important point to be noted is that, two statistical techniques namely permutation entropy & residual skewness approach discussed in Section 4.3 was implemented using a python based open source library namely EntroPy[4] & Scipy[5] respectively.

6.2. Data collection & pre-processing

In SWaT testbed, each sensor measurements and actuator's status are stored in historian at one second granularity. For testing purpose, we utilize the records from historian, while the plant is operating in steady state. Further, records extracted from the historian are converted to the compatible format for the considered I-DCNN. As each records needs to be represented in a numeric format, the status of actuators are converted to "2" in case of "OPEN" or "ON" and "1" in case of "CLOSE" or "OFF". The transient state of actuators is also represented as "1".

[4] https://raphaelvallat.com/entropy/build/html/index.html
[5] https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.skew.html

6.3. Parameters fine tuning

During the initial phase of the experiment, we build independent models for each sensors of Stage 1-3 namely FIT101, LIT101, FIT201, and LIT301. The Table 2 shows the dependency of considered sensors with other SWaT components, extracted from invariants [5]. As discussed in the Section 3.2, the Eqn. (5) represent a non-linear relationship among the state variables with single order time delay. However in real time, opening or closing of valve does not happens immediately when the control signals are initiated by PLCs. Instead their exists a transient state that incurs few seconds so that, we experience 11 % increase in the inflow rate of water (FIT sensor value) during the normal SWaT operation. Although, we neglect the existence of transient state in this study, but their time delay has to be accounted during the model creation in order to minimize the false positives. Thus Eqn. (5) can be modified as follows,

$$s(t+1) = f(s(t), \phi_{in}(t_k), \phi_{out}(t_k)) \tag{22}$$

Where t_k corresponds to the time delay due to the transient state of actuators. Along with t_k, we identify the optimal value of CNN's hyper-parameters ie number of filters in the convolution layer, number of hidden units in fully connected layer etc using the dataset obtained from [14]. The Table 3 shows the parameter setting of the considered CNN architecture for the design of I-DCNN. Generally, the seven days of records representing the normal operation of SWaT is divided in a ratio of 80:20 for training and validation purposes.

The training process was carried out on a three distinct CNN architecture with varying set of convolution & pooling layer namely (i) $DCNN_1$ (CNN with one set of convolution & pooling layer),(ii) $DCNN_2$ (CNN with two set of convolution & pooling layer), (iii) $DCNN_3$ (CNN with three set of convolution & pooling layer). Finally as shown in Figure 10, the best architecture with fine tuned hyper-parameters is obtained through the P-value computed from Eqn.(11). Subsequently in the validation process, around 120,960 records were utilized to identify the parameter of various statistical techniques discussed in Section 4.3. Further dataset used for training and validation process represents the behaviour

Table 2
Invariant for each sensor in Stage 1-3

Invariant	Dependent variable	Independent variable
I1	FIT101(t)	FIT101(t-1),MV101(t-t_k)
I2	LIT101(t)	LIT101(t-1),FIT101(t-t_k), FIT201(t-t_k)
I3	FIT201(t)	FIT201(t-1),P101(t-t_k), MV201(t-t_k)
I4	LIT301(t)	LIT301(t-1), FIT201(t-t_k), FIT301(t-t_k)

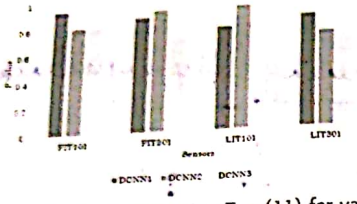

Figure 10: P-value computed using Eqn.(11) for various CNN architectures

Table 3
Hyperparameter values of I-DCNN

S.No	Hyperparameter	Range
1	Filters in convolution layer	[32,64,128]
2	Hidden units in fully connected layer	[50,100,150, 200,250,300]
3	t_k	[2-10]
4	Number of epochs	[10,20,50]
5	Batch size	[32,34]

of the plant in late 2015. Over a period of time, ie., 4 years, there has been several improvements and maintenance work carried out. For example, replacement of faulty components, change in the control codes of PLCs, positioning new sensors & actuators etc. Before analysing the performance of the proposed anomaly detector, it is mandatory to ensure the reliability of the computed statistical parameters. Hence, the plant was operated for 2 hours in normal state to fine tune the statistical parameters with the current behaviour.

6.4. Performance analysis

The testing phase of the proposed anomaly detector was carried out on the operational plant under two scenarios. One with the normal operation to compare the variants of I-DCNN in terms of false positives and another with several attacks launched against the SWaT components, for comparison in terms of accurate & faster detection of stealthy attacks. In the entire testing process, monitoring the system behaviour and anomaly detection was carried out using the live data recorded in data historian.

In the initial scenario, the plant was operated in a steady state without any launch of attacks for two hours. During this process, none of the I-DCNN variants for four different sensors raised false alarms. This is due to the fine tuning of parameters (discusses in Section 6.3) with both historical and live data from SWaT testbed. The existence of zero false positives in the variants of I-DCNN implies that the physical model designed using the invariants along with the integrated statistical techniques aligned themselves with the normal system behaviour of the SWaT testbed. In the next scenario, the performance of I-DCNN was analysed by launching several attacks discussed in Section 5. As we compare the variants of proposed anomaly detector in terms of detection time, each attack was launched until the intended objective (discussed in Table 1) of the corresponding attacks are realised. In order to avoid the physical damage of components like pumps & valves, few alterations are done in the water level markers of tank T101 and T301 such that, overflow or underflow does not happen physically. In the below discussion, we claim that an attack is detected only when the alert is raised by the corresponding model designed for a particular sensor.

As indicated in Table 4, all single (A1 - A12) & multi-point (A13-A16) attacks launched against the SWaT components were detected by DCNN integrated with CUSUM & GD approach's before the intention is realised. However, other two approaches does not exhibit the similar performance. Out of 60 single point attacks, DCNN-PE & DCNN-RS detects only 22 & 35 attacks respectively. Similarly in case of multi-point attacks, out of 20 launched in stage 1 - 3 of SWaT testbed , DCNN-PE detects only 4 attacks whereas DCNN-RS detects 9 attacks. Unlike CUSUM & GD, other two are non-parametric approaches and rely on the randomness contained in the residual sequence. To minimize false positives in DCNN-PE, a time interval to which the $H(\lambda)$ (Eqn.(15)) can increase or decrease is computed during the validation process. Since the attacks are stealthy in nature, the time interval is to large so that the attacks are not detected effectively in the testing phase. Although, RS approach has a similar strategy like PE approach, but its better performance is due the existence of parameter 'τ' (Section 4.3). Further, as Eqn(17) is computed based on the skewness of the residual sequence, limits its performance similar to CUSUM & GD approach's. In the following discussion, we compare the effectiveness of DCNN integrated with CUSUM & GD approaches in terms of detection time.

Apart from single & multi-point, we can categorize the attacks in Table 1, as "surge" & "bias" on basis of their nature. In surge attack, the attacker tries to achieve maximum damage to physical components as soon as possible and in bias attack, the attacker injects a small value over a longer period of time. However, in both cases the magnitude in the difference between the actual & mutated value will be minimal. The attacks A1,A3,A7 & A9 falls in surge type and remaining attacks are of bias in nature. From the Figure 11, DCNN-CUSUM approach detects both bias & surge attacks in a faster rate compared with DCNN-GD approach. As discussed in Section 4.3, the CUSUM approach operates on the rate in change of error in residual sequence whereas

Table 4
Effectiveness of I-DCNN in detecting process anomalies against SWaT
(* Single point attack; ** Multi-point attack)

Attack Reference	No. of attacks launched	I-DCNN variants			
		DCNN-CUSUM	DCNN-GD	DCNN-PE	DCNN-RS
A1*	5	5	5	3	4
A2*	5	5	5	2	3
A3*	5	5	5	1	3
A4*	5	5	5	1	3
A5*	5	5	5	2	2
A6*	5	5	5	0	1
A7*	5	5	5	3	4
A8*	5	5	5	3	4
A9*	5	5	5	2	3
A10*	5	5	5	1	3
A11*	5	5	5	3	2
A12*	5	5	5	1	3
A13**	5	5	5	1	3
A14**	5	5	5	1	2
A15**	5	5	5	1	2
A16**	5	5	5	1	2

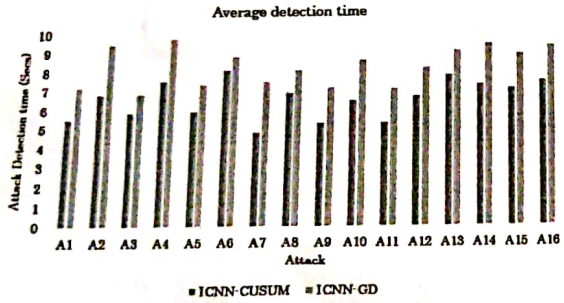

Figure 11: Average detection time of stealthy attacks

the GD approach rely on the Gaussian distribution. Thus it can detect even a minor deviation in the residues that corresponds to anomalies. An important point to be noted from Figure 11 is that, the detection time of bias attacks using both approaches are higher compared with the time for detecting surge attacks. This is due to the nature of bias attack. As the attacker injects a error value with minimal magnitude, it incurs few seconds for the to detect it existence in the residual sequence.

6.5. Research Questions and Contributions

The research questions formulated in Section 1 are revisited in context with the experimental results discussed above.

RQ1: *How to effectively model the physical process of ICS using Deep convolutional neural network?*

As discussed in Section 3.2, each process in the underlying ICS are based on the fundamental laws of physics [9]. A formal physical model can be derived from these fundamental laws along with the availability of operational data. For the considered water treatment plant, the dependency of each components are derived inform of invariants through application of law of physics, by the author of [5]. In this work, we attempt to capture the relationship through the deep convolutional neural network through the application of historical data available in [14] & from the operational SWaT testbed. Further, the performance of DCNN is improved by fine tuning their values of hyperparameters based on the current system behaviour, so that the higher order of non-linear relationship among the state variables are captured effectively resulting in the prediction of expected system behavior with minimal forecasting error. Observations from the Figure 10, reveals the significance of I-DCNN in efficiently analyzing and predicting the behavior of SWaT components.

RQ(2): *How fast and accurately detect the stealthy attacks before its objective is achieved?*

The accurate and faster detection of stealthy attacks against ICS mainly rely on two factors, (i) technical knowledge of the attacker (ii) strength of the defence mechanism. As discussed in Section 5, we assume that attacker is of insider type and has prior knowledge regarding each process and the specifications of SWaT components. Thus in this reported study, the strength of the defence mechanism plays a significant role in the effective detection of stealthy attacks.

The proposed anomaly detector consists of two phases. Firstly using the dependency among the SWaT components, a physical model is built using DCNN to predict the expected behaviour of the system with minimal error. Secondly, using several statistical techniques discussed in Section 4.3, the predicted behaviour is compared against the actual behaviour for anomaly detection. From the comparative study (Section 6.4), we can conclude that DCNN integrated with CUSUM approach detects the stealthy attacks at faster rate with better detection accuracy compared with other statistical techniques.

In stealthy attacks, the attacker will inject the false measurements closer to the expected behaviour of the system for a longer time to achieve their goal and remains undetected. The proposed anomaly detector based on CUSUM approach proves to be better solution for the detection stealthy attacks in SWaT testbed due to following reasons: (i) CUSUM approach tracks the rate of change of residual error(no matter how minimal it is) for anomaly detection (ii) The parameter of CUSUM (target & b) discussed in Section 4.3, are fine tuned based on normal behaviour of plant using both historical and live data. (iii) The upper & lower control limit (Eqn.(12 & 13)) was empirically computed to minimize the false alarms due to sensor noise and temporal glitches.

RQ(3): *Is there a possibility of locating the exact ICS component under threat??* Generally, the anomaly detectors designed for ICS are either univariate or multivariate in nature. In former case, a representative model is built by the temporal dependency of the single sensor or actuator, whereas in latter case, the process models are derived from the law of physics ie the interaction among the components. The proposed anomaly detector is of multivariate in nature. Although, the design of univariate based anomaly detectors are easy and computationally attractive, they fails to detect multi-point & coordinated attacks with are more common in ICS.

The main reason behind is that, they are based on the behaviour of single component. However, in case of I-DCNN, a multi-variate anomaly detector, possess the ability to effective detection of multi-point & coordinated attacks, which is evident from Section 6.4. In addition, I-DCNN enables the plant operator or administrator to virtualize the current and predicted behaviour of each components in real time and identify the attack location.

6.6. Generality & limitations of I-DCNN

The generation of invariants and methodology used to create models for I-DCNN is only not limited to SWaT testbed but can be extended to other ICS like water distribution plant, power grids, etc. For example the authors of [11] have discussed the methods to generate invariants for Water distribution plant (WADI). As the extension of this work, with the dataset consisting of all operational modes of WADI plant available in [1], several models of I-DCNN can be designed to monitor each process in WADI testbed. One of the significant difference in application of I-DCNN in SWaT & other testbeds is the optimal values of hyper parameters computed based on the behaviour of each state variable discussed in Section 6.3. Unless the proposed anomaly detector deployed and validated on other realistic plant, we cannot comment on its generality.

6.7. Conclusion

Requesting Prof Aditya to write

[2] Adepu, S., Mathur, A., 2016a. An investigation int water treatment system to cyber attacks, in: 2016 tional Symposium on High Assurance Systems Er IEEE. pp. 141–148.

[3] Adepu, S., Mathur, A., 2016b. Using process inv ber attacks on a water treatment system, in: IFIP I ence on ICT Systems Security and Privacy Prot 91–104.

[4] Adepu, S., Mathur, A., 2018a. Assessing the e detection at a hackfest on industrial control sys tions on Sustainable Computing .

[5] Adepu, S., Mathur, A., 2018b. Distributed att ter treatment plant: method and case study. Dependable and Secure Computing .

[6] Adepu, S., Mathur, A.P., 2016c. Detecting n water treatment system using intermittent co CRC, pp. 59–74.

[7] Asghar, M.R., Hu, Q., Zeadally, S., 2019. trial control systems: Issues, technologies, ar Networks 165, 106946.

[8] Bhamare, D., Zolanvari, M., Erbad, A., Jai N., 2019. Cybersecurity for industrial co Computers & Security , 101677.

[9] Cárdenas, A.A., Amin, S., Lin, Z.S., Hu Sastry, S., 2011. Attacks against process sessment, detection, and response, in: Pr symposium on information, computer and ACM. pp. 355–366.

[10] Cárdenas, A.A., Amin, S., Sastry, S., 200 the security of control systems., in: HotS

[11] Feng, C., Palleti, V.R., Mathur, A., Cha framework to generate invariants for an control systems., in: NDSS.

[12] Gamage, T.T., McMillin, B.M., Roth, T tion flow security properties in cyber- ized framework based on compensatio Computer Software and Applications pp. 158–163.

[13] Gauthama Raman, M.R., Somu, N., M detection in critical infrastructure usin in: Shankar Sriram, V.S., Subramaniy Zhang, L., Batten, L., Li, G. (Eds.), A Information Security, Springer Singap

[14] Goh, J., Adepu, S., Junejo, K.N., M support research in the design of secu International Conference on Critical curity, Springer. pp. 88–99.

[15] Heng, J., Huei, Y.C., 2019. Mach anomalies in secure water treatmen ence on Machine Learning and Co 2019), pp. 129–136.

[16] Hu, Y., Li, H., Luan, T.H., Yang 2018. Detecting stealthy attacks a permutation entropy-based me Systems .

[17] Hu, Y., Li, H., Yang, H., Sun, Y stealthy attacks against industr skewness analysis. EURASIP and Networking 2019, 74.

[18] Inoue, J., Yamagata, Y., Ch Anomaly detection for a wate machine learning, in: 2017 I Mining Workshops (ICDMW

[19] Jagtap, S.S., Shankar Srira

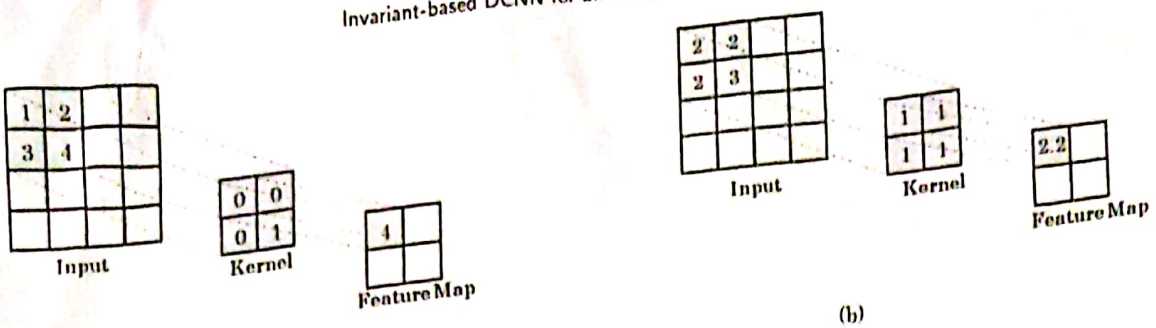

Figure 5: Pooling operation in CNN: (a) Max pooling; (b) Average pooling

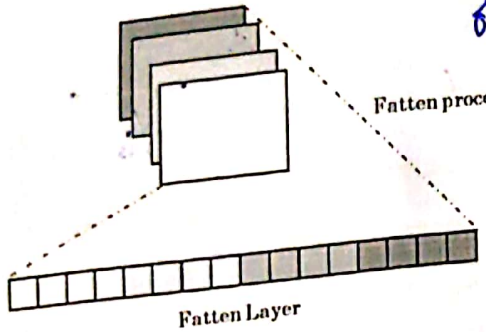

Figure 6: Flatten operation in CNN

to adjcent layers for proving linear output defined in Eqn.(9) for regression problems

$$Y_{fc} = \varphi((W_{fc} * P) + b_{fc}) \quad (9)$$

where Y_{fc} is the overall output of CNN, W_{fc} & b_{fc} are the weight matrix and bias of fully connected layer, respectively.

4.2. Hyperparameters Fine tuning

The outstanding performance of several variants of CNN like AlexNet, ResNet, VGG, GoogleNet, DenseNet etc, in variety of applications is not only due to the intellectual design of CNN's architecture, but also the proper selection of hyper-parameters value. In case of regression problems, these hyper-parameters are fine tuned with a constrain to minimize the prediction error as given in Eqn.(10)

$$Minimize\ f_e = \frac{1}{n}\sum_{i=1}^{n}(s_i(t) - \hat{s}_i(t))^2; w.r.t H_p \quad (10)$$

where f_e is the forecasting error, $s_i(t)$ & $\hat{s}_i(t)$ are the actual & predicted state of i^{th} component at t^{th} timestep and H_p is the hyper-parameters.

In case of CNN, the problem of fine tuning the hyper-parameters can be viewed as a multi modal function optimization as the hyper-parameters includes parameters in convolution layer (kernal size number, padding), pooling layer (pooling type, padding) and fully-connected layer (number of neurons layers, activation function). Thus the manual tuning is a costly trail-and-error way. Further, SWaT testbed is of heterogeneous nature and generates a massive volume of multivariate data with non-linear relationship, making the manual process a difficult task. Hence in this work, we adopt automated grid search algorithm for fine tuning, where multiple CNN models with different values of hyperparameters are created and best model is selected using the *P-value* of Kolmogorov-Smirnov test (K-S test) defined in Eqn.(11).

$$K_s = \sup_t |s(t) - \hat{s}(t)| \quad (11)$$

where sup, is the supreme of the set of distances. The *P-value* of K-S test will always be in the interval [0,1], where 1 indicates $s(t)$ & $\hat{s}(t)$ are similar.

4.3. Residual error-based anomaly score

For a given invariant modelled through DCNN, whose hyper-parameters are fined tuned during the training process, the predicted state of a component is compared against its observed state for anomaly detection. In order to ensure the normal operation of the plant, the computed residue (r(t)) must be within an acceptable range (τ & t_w) as discussed in Section 3.3. However, due to several factors like non-linearity, aging of components, misconfiguration, temporal glitches etc, of the operational plant, there exists a significant challenge in computing the optimal values of τ and t_w. Moreover, the anomaly detection process is more sensitive to these values and their poor choice will lead to high number of false alarms.

According to Yan Hu et al. [16] the residual series generated during the normal operation of the plant will be of random in nature and similarly during an attack scenario there exists some sort of non-random dynamics in it. Through the application of statistical approaches, these patterns or regularities can be characterized for effective detection of process anomalies. In this reported study, the following statistical techniques are integrated with DCNN for monitoring